

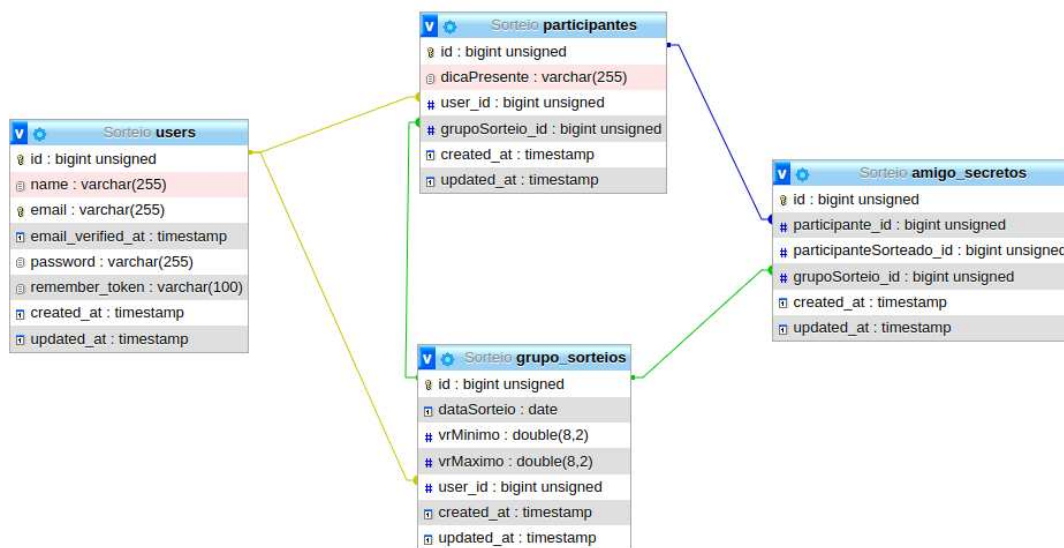
1 Introdução

Para exemplificar o acesso e a utilização de dados da autenticação, será criado um projeto para simular a criação de grupos para sorteio de um amigo secreto.

Cada usuário poderá criar um grupo, onde os demais usuários poderão se inscrever, e após a realização do sorteio, verificar quem foi seu amigo secreto sorteado.

Apenas o criador do grupo poderá: Realizar um sorteio, desfazer um sorteio realizado, editar ou excluir o grupo de sorteio.

O modelo de entidade e relacionamento para este projeto foi definido assim:



- **users:** Usuários do sistema → tabela padrão do módulo de autenticação do Laravel.
- **grupo_sorteios:** Armazenará os grupos criados. Possui uma data, valor mínimo, valor máximo, e o id do usuário que criou.
- **participantes:** Armazenará o id do usuário e o id do grupo sorteio que está inscrito e a dica de presente.
- **amigo_secretos:** Armazenará os dados do sorteio em si. É a tabela que estabelece o relacionamento entre *participantes* e *grupo_sorteios*. Tem o id do grupo, o id do participante e o id do seu amigo sorteado.

2 O Projeto: Sorteio

- `composer create-project laravel/laravel Sorteio`
- `npm install`
- `composer require laravel/ui --auth`
- `php artisan ui bootstrap`
- `npm install jquery --save-dev`
- `npm install`
- `npm run dev`
- alterar os arquivos: `vite.config.js`, `resources\js\bootstrap.js` e `resources\js\app.js`, conforme já visto em exemplos anteriores.

3 Camada Model

Conforme o diagrama apresentado na seção 1, teremos os seguintes Models:

- `User.php` → criado pelo Laravel ao habilitarmos o módulo de autenticação
- `GrupoSorteio.php`
- `Participante.php`
- `AmigoSecreto.php`

4 Camada Controller

Foram criados dois controllers:

- **`GrupoSorteioController.php`**
 - `construct()`: carrega o middleware de autenticação.
 - `index()`: carrega os grupos de sorteio criados. Verifica se o "sorteio" já foi realizado e se o usuário autenticado é participante deste grupo. Envia os dados para a view *sorteios*.
 - `create()`: retorna a view para criação de um novo grupo (*novoSorteio*).
 - `store()`: grava o novo grupo criado e redireciona para que o criador do grupo seja incluído como participante.
 - `edit()`: carrega os dados do grupo para serem editados na view *editarSorteio*.
 - `update()`: atualiza os dados do grupo no BD.
 - `destroy()`: remove o grupo do BD.
 - `sortear()`: utilizando o recurso de geração de números aleatórios, realiza o "sorteio" entre os participantes inscritos no grupo e grava na tabela *amigo_secretos*.
 - `deletarSorteio()`: remove os registros da tabela *amigo_secretos*, desfazendo o sorteio realizado.

- **ParticipanteController.php**

- *construct()*: carrega o middleware de autenticação.
- *index()*: carrega os participantes de um grupo e envia para a view *listaParticipantes*.
- *create()*: verifica se o usuário já é participante do grupo, se não, carrega a view *novaInscricao*; caso contrário, envia uma mensagem de que o usuário já está inscrito naquele grupo.
- *store()*: grava os dados na tabela *participantes*.
- *edit()*: carrega a view *editarPresente*.
- *update()*: atualiza os dados na tabela *participantes*.
- *destroy()*: remove o participante de um grupo.
- *verAmigo()*: caso o sorteio já tenha sido realizado, permite ao usuário visualizar quem foi seu "amigo secreto" sorteado.

5 Camada View

- **welcome e home**: views geradas pelo Laravel e adaptadas para este projeto.
- **sorteios**: apresentará os grupos de sorteio existentes e uma opção para criar um novo grupo.
- **novoSorteio**: formulário para criar um novo grupo.
- **editarSorteio**: formulário para editar os dados de um grupo.
- **listaParticipantes**: apresenta os participantes inscritos em um grupo.
- **novaInscricao**: formulário para inscrever-se em um grupo.
- **editarPresente**: formulário para editar a dica de presente cadastrada em um grupo.
- **verAmigo**: apresenta o "amigo secreto" sorteado.

6 Rotas: routes/web.php

A rota "raiz" direciona para a view *welcome*, e as demais rotas necessárias para cada um dos métodos dos controllers: *GrupoSorteioController* e *ParticipanteController*.

7 Link para os arquivos do projeto

- <https://github.com/MarceloMussel/ArquivosProjetoAmigoSecreto>